

MODUL PRAKTIKUM IX

EXCEPTION HANDLING

A. Tujuan

1. Mahasiswa mampu menganalisa dan menerapkan penanganan kesalahan program dengan *exception handling*.
2. Mahasiswa mampu membuat class *exception* sendiri.

B. Latihan

Latihan 1:

Implementasikan *source code* berikut :

ThrowDemo.java

```
public class ThrowDemo {  
  
    public static void main(String args[]) {  
        String input = "invalid input";  
        try {  
            if (input.equals("invalid input")) {  
                throw new RuntimeException("throw demo");  
            } else {  
                System.out.println(input);  
            }  
            System.out.println("After throwing");  
        } catch (RuntimeException e) {  
            System.out.println("Exception caught here.");  
            System.out.println(e);  
        }  
    }  
}
```

Output dari program di atas adalah:

Latihan 2:

Implementasikan *source code* berikut :

1. Barang.java

```
public class Barang {  
  
    private String kode;  
    private String nama;  
    private double harga;
```

```

public void setKode(String vKode) {
    try {
        kode = vKode;
        if (kode == null) {
            throw new NullPointerException();
        }
    } catch (NullPointerException npe) {
        System.out.println("KESALAHAN: "
            + "Kode barang tidak boleh null");
    }
}

public String getKode() {
    return kode;
}

public void setNama(String vNama) {
    try {
        nama = vNama;
        if (nama == null) {
            throw new NullPointerException();
        }
    } catch (NullPointerException npe) {
        System.out.println("KESALAHAN: "
            + "Nama barang tidak boleh null");
    }
}

public String getNama() {
    return nama;
}

public void setHarga(int vHarga) {
    harga = vHarga;
}

public double getHarga() {
    return harga;
}
}

```

2. DemoThrow.java

```
public class DemoThrow {  
    public static void main(String[] args) {  
        Barang obj = new Barang();  
  
        obj.setKode(null);  
        obj.setNama("Buku tulis");  
        obj.setHarga(2500);  
  
        System.out.println("\nKode : " + obj.getKode());  
        System.out.println("Nama : " + obj.getNama());  
        System.out.println("Harga : " + obj.getHarga());  
    }  
}
```

Output dari program di atas adalah:

Latihan 3:

Implementasikan *source code* berikut :

ThrowDemo2.java

```
public static void test(int n) {  
    if (n < 0) {  
        throw new NullPointerException(  
            "KESALAHAN: null pointer");  
    } else {  
        throw new ArithmeticException(  
            "KESALAHAN: arithmetic exception");  
    }  
}  
  
public static void main(String[] args) {  
    try {  
        test(-12); // menimbulkan eksepsi NullPointerException  
  
        test(0); // menimbulkan eksepsi ArithmeticException  
    } catch (Exception e) {  
        System.out.println("Eksepsi ditangkap di sini...");  
        System.out.println(e.getMessage());  
    }  
    System.out.println("Statement setelah blok try-catch");  
}
```

Output dari program di atas adalah:

Latihan 4:

Implementasikan *source code* berikut :

1. NegativeNumberException.java

```
public class NegativeNumberException extends Exception {  
  
    private int bilangan;  
    // default constructor  
    NegativeNumberException() {  
    }  
    // constructor dengan parameter bertipe String  
    NegativeNumberException(String pesan) {  
        super(pesan);  
    }  
    // constructor dengan parameter bertipe String dan int  
    NegativeNumberException(String pesan, int nilai) {  
        super(pesan);  
        bilangan = nilai;  
    }  
    public int getBilangan() {  
        return bilangan;  
    }  
}
```

2. CreateNewException.java

```
public class CreateNewException {
    public static int hitungFaktorial(int n)
        throws NegativeNumberException {
        if (n < 0) {
            throw new NegativeNumberException(
                "Bilangan tidak boleh negatif", n);
        }
        int hasil = 1;
        for (int i = n; i >= 1; i--) hasil *= i;
        return hasil;
    }
    public static void main(String[] args) {
        System.out.println("Pada saat menghitung 5!");
        try {
            System.out.println("Hasil = " + hitungFaktorial(5));
        } catch (NegativeNumberException nne) {
            System.out.println("Bilangan : " + nne.getBilangan());
            nne.printStackTrace();
        }
        System.out.println("\nPada saat menghitung -5!");
        try {
            System.out.println("Hasil = " + hitungFaktorial(-5));
        } catch (NegativeNumberException nne) {
            System.out.println("Bilangan : " + nne.getBilangan());
            nne.printStackTrace();
        }
    }
}
```

Output dari program di atas adalah:

C. Tugas Rumah

Modifikasi 2 (dua) program yang sudah pernah kalian buat pada praktikum sebelumnya. Tambahkan *exception handling* yang berbeda pada kedua program tersebut. Uji coba dengan menggunakan kondisi normal dan kondisi yang memicu terjadinya *exception*.